Shape and Pose Recovery from Planar Pushing

Kuan-Ting Yu[1], John Leonard[1,2], Alberto Rodriguez[2]
[1] Computer Science and Artificial Intelligence Laboratory — Massachusetts Institute of Technology
[2] Mechanical Engineering Department — Massachusetts Institute of Technology
<peterkty,jleonard>@csail.mit.edu, albertor@mit.edu

***Abstract*— Tactile exploration refers to the use of physical interaction to infer object properties. In this work, we study the feasibility of recovering the shape and pose of a movable object from observing a series of contacts. In particular, we approach the problem of estimating the shape and trajectory of a planar object lying on a frictional surface, and being pushed by a frictional probe. The probe, when in contact with the object, makes observations of the location of contact and the contact normal.**

Our approach draws inspiration from the SLAM problem, where noisy observations of the location of landmarks are used to reconstruct and locate a static environment. In tactile exploration, analogously, we can think of the object as a rigid but moving environment, and of the pusher as a sensor that reports contact points on the boundary of the object.

A key challenge to tactile exploration is that, unlike visual feedback, sensing by touch is intrusive in nature. The object moves by the action of sensing. In the 2D version of the problem that we study in this paper, the well understood mechanics of planar frictional pushing provides a motion model that plays the role of odometry. The conjecture we investigate in this paper is whether the models of frictional pushing are sufficiently descriptive to simultaneously estimate the shape and pose of an object from the cumulative effect of a sequence of pushes.

I. Introduction

Tactile feedback complements our global view of the world with local but detailed information about mechanical properties such as roughness, compliance, pose, or shape [1]. Humans, for example, can search very efficiently through the insides of a bag when blindfolded. We have little problem in identifying a familiar object, or even delineating its shape from tactile cues [2].

The last two decades have seen a wealth of sensing technologies aimed at recovering tactile information, with substantial improvements in resilience, sensitivity, and ease of integration. The question of how to make an efficient and timely use of that information still remains a challenge.

The end-goal of our work is a broad capability for tactile inference. In this paper, we study how feasible it is to understand the pose and the shape of a movable planar object through tactile interactions. The planar object lies unconstrained, save for friction, on a flat surface, and a probe pushes it while exploring its contour, as in Figure 1.

Our approach relies on a model of the mechanics of frictional sliding [3] to predict the motion of the pushed object. The main question we explore is whether this model, in combination with contact information, is sufficient to recover the trajectory and shape of the object. We make the following assumptions:

- Out-of-plane effects are negligible;
- a quasi-static pushing process;
- object and table interact via a frictional planar contact with uniform friction and known pressure distribution;
- object and pusher interact via a frictional point contact.
- the probe can reach all points on the surface of the object, and when in contact, the location of contact and contact normal are well defined, except maybe at corners.

Our contributions include:

- a formulation of the problem of tactile reconstruction;
- an approach to solve it;
- a metric to evaluate the fitness of a solution;
- a study of the sensitivity of the solution to imperfections in the motion and observation models.

Our approach starts from a given sequence of contact points and contact normals between object and probe. For the purpose of this paper we gather the data with a simple contour following strategy. We then use a Bayesian model to fit the contact observations to model the motion of the object. More specifically, we form a belief network in poses, observations, and shapes, with which we find a maximum a posteriori estimate for pose and shape, in the spirit of the

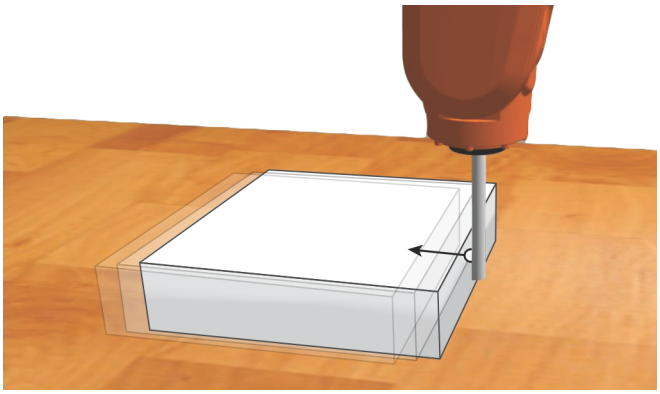

Fig. 1. Setup for planar tactile exploration: A robotic manipulator moves a vertical probe that pushes a flat object on a support surface. The pusher observes both the contact location and the contact normal with the object.

This work was supported by NSF award [NSF-IIS-1427050] through the National Robotics Initiative.

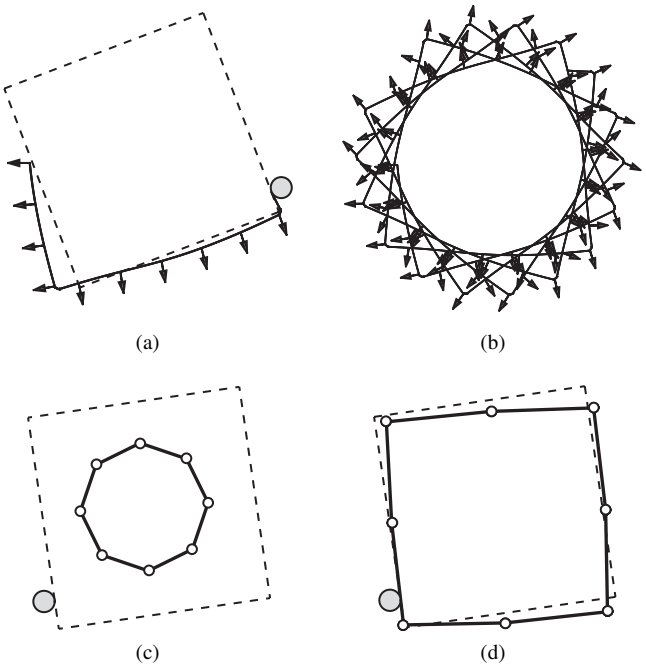

Fig. 2. Illustration of the tactile reconstruction problem. (a) Typical input data—contact locations and contact normals—when exploring a fraction of the object boundary; (b) Input data for four complete revolutions of the pusher around the object boundary; (c) Example of initial guess we input to the optimizer; (d) Final estimated shape and pose of the object for an instant of the trajectory. The solid line and white circles show the estimated shape, and the dashed line ground truth.

conventional GraphSLAM algorithm [4]. In this problem, the motion model is defined by the prediction of frictional pushing, and the measurement model observes contact position and contact normal.

Finally, we formulate an objective function from the belief network, which we optimize in batch subject to the constraints from the motion and observation models. Figure 2 shows a typical example of the tactile data gathered from pushing an object, a typical initial guess we input to the optimizer, and an instant of the resultant estimated shape and pose of the object. To the best of our knowledge, this is the first work that addresses both pose and shape recovery in the context of pushing.

The paper is organized as follows:

I. Introduction;
II. related work;
III. problem formulation;
IV. review on mechanics of frictional pushing;
V. simulated experiments;
VI. discussion of results and limitations;
VII. future work.

II. Related Work

This paper describes a least-squares approach to recovering the shape and trajectory of an object from tactile interaction. It builds on an understanding of the mechanics of planar pushing and sliding.

Mason [5] proposed one of the earliest models for the mechanics of pushing. He described an algorithmic resolution technique in the form of a "voting" theorem to determine the direction of rotation of a pushed object: clockwise, counterclockwise, or straight. Mason also pointed out the challenges in accurately predicting the motion of a pushed object. The indeterminacy of the pressure distribution between object and its support, leads often to incorrect assumptions on the location of the center of pressure and center of rotation.

Under the assumption of a known pressure distribution, Goyal et al. [3] characterized the set of all possible frictional forces on a planar sliding object with a geometric construction, the *limit surface*. They used it to efficiently compute the motion of a sliding object subject to an external load [6].

Lee and Cutkosky [7] generalized the concept of limit surface for uneven pressure distributions, and proposed three different methods to compute it within the context of fixture design for automation. Particularly relevant to this paper is their ellipsoidal approximation, which provides a compact and easily invertible relationship between motions and forces. Lynch et al. [8] used it to construct an analytical solution to the motion of a pushed object under the assumption of quasi-static interaction. In this paper, we use that approximation as the motion model of our Bayesian approach to reconstructing the shape and trajectory of the object.

Contact-based shape reconstruction. The closest work to this paper is by Moll and Erdmann [9] who explored the problem of reconstructing both the shape and pose of a convex smooth object by rolling it in between two planar palms. Previously, Jia and Erdmann [10] had constructed observers to estimate the pose and predict the motion of an object dynamically pushed.

Petrovskaya and Khatib [11] applied a particle filter to the problem of estimating the 6 DOF pose of an object rigidly attached to a robot arm, and proposed the use of *scaling series* to improve the efficiency of the estimation. More recently Martinez-Hernandez et al. [12] demonstrated a contour following technique to reconstruct the 2D shape of an object also rigidly attached to a table. Strub et al. [13] approached the same problem for an object pinned to a table, limiting its mobility to a single rotational freedom. They show how to actively use the rotation of the object and tactile sensing to recover its shape. Our work relaxes completely the constraints on the mobility of the object, except for frictional interaction with the table.

Also closely related to contact-aware state estimation, Zhang et al. [14] used a particle filter to estimate the location of an object and dynamic properties such as dimension, mass, and friction. More recently, Koval et al. [15] introduced the *manifold particle filter*, imposing a hard constraint on the binary contact/no-contact condition when tracking the pose of an object.

Vision-based shape reconstruction. The problem of shape reconstruction from a moving camera has been extensively explored in the vision community. The problem is known as *structure from motion* [16]. Large-scale 3D shape reconstruction has been demonstrated from multiview 2D images [17], with Lidar pointclouds [18], or more recently from RGB-

D images [19]. Although the scenarios in vision-based and touch-based sensing are different, they share a similar structure in terms of information extraction and inference. The process is discussed further in the following section.

III. Problem Formulation: Shape and Pose Recovery

We are concerned with the problem of simultaneously estimating the shape and the trajectory of a rigid 2D object pushed on a table.

Let $\mathcal{O}$ be an object in the plane, with position and orientation denoted by $q = (x, y, \theta)$, and boundary determined by a series of control points $\{s_i\}_{1\dots N}$. We denote $\vec{r}_p$ as the unique contact point between pusher and object, and $\hat{n}$ to be the corresponding contact normal. The interaction between object and pusher is observed with periodicity, and we use the subscript $t \in [1...T]$ to indicate the corresponding timestamp along the trajectory. We formulate the problem of tactile reconstruction as:

Problem 1 (Tactile Reconstruction): Let $\{\vec{r}_{p_t}, \hat{n}_t\}_{1\dots T}$ be a series of contact locations and contact normals of a probe with a movable planar object $\mathcal{O}$. The object is subject to surface friction with the plane where it slides, and point friction with the pusher. Estimate the control points $\{s_i\}_{1\dots N}$ that determine the boundary of $\mathcal{O}$, and the series of poses $\{(x_t, y_t, \theta_t)\}_{1\dots T}$ that determine its trajectory.

A. The Approach

Our approach formulates the joint estimation of shape and trajectory as a least-squares optimization problem by assuming Gaussian noise both in the contact measurements and the motion predictions. We construct a cost function that captures the fit of the estimated shape and trajectory to the predicted motions and observations of the object. Parallel to the SLAM problem [20], the optimization has the form of a belief network, as illustrated in Figure 3, and the solution has the form of a *maximum a posteriori* (MAP) estimate.

A measurement $\mathbf{z}_t$ includes the observed contact location $\vec{r}_{p_t}$ and the corresponding contact normal $\hat{n}_t$. For the sake of simplicity, we assume that pusher and object are always in contact, so that $\mathbf{z}_t$ is continuously well defined.

The optimization takes place in the space of the control points of the object shape $\{s_i\}_{1\dots N}$, and the control points of its trajectory $\{(x_t, y_t, \theta_t)\}_{1\dots T}$. We aggregate all these decision variables into a large vector X, and define a cost function:

$$E(X) = [|D(X)|^2, |G(X)|^2, |K(X)|^2, |L(X)|^2, |S(X)|^2]\boldsymbol{\alpha} \quad (1)$$

composed of five terms that capture:

- D, G: fit of a solution $\hat{X}$ to the observed contact locations and contact normals;
- K, L: fit of a solution $\hat{X}$ to the predictions of the motion model in linear and angular displacements;
- S: shape prior that encourages control points to be equally spaced.

The belief network in Figure 3 shows the relationship between these constraints. In the current form of this work, we hand picked their weighting $\boldsymbol{\alpha}$. A more mature implementation would tune the weights based on their information matrix [4], to put more trust on constraints with higher confidence.

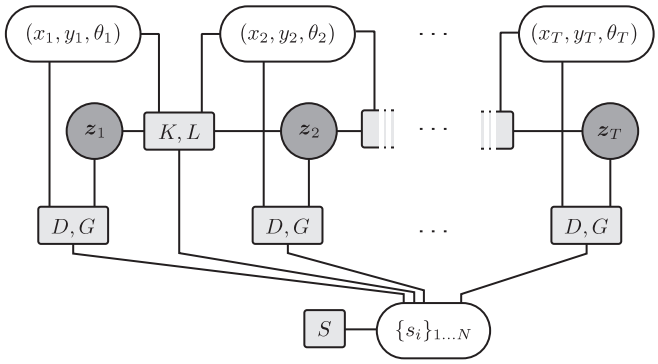

Fig. 3. Graphical representation (factor graph) of the cost function to optimize. The dark shaded circles are the observations $\mathbf{z}_t = (\vec{p}_t, \hat{n}_t)$. The factors D, G, K, L, and S in light shaded squares, constraint the values of the states of the problem, represented in white background, that include the shape of the object and its series of poses.

Measurement Cost. Terms D and G enforce that the measured contact points $\{\vec{r}_{p_t}\}_{1\dots T}$ lie on the boundary of the object, and that the normals $\{\hat{n}_t\}_{1\dots T}$ point in the right direction.

More precisely, for each instant t, D_t is the distance between the observed contact point $\vec{r}_{p_t}$, and the closest point to the estimated boundary of the object. We approximate the object's boundary by a polygon with vertices at the control points $\{s_i\}_{1\dots N}$. G_t, in turn, is the distance between the measured contact normal $\hat{n}_t$ and the normal at the closest point to the estimated boundary of the object.

Note that in the conventional SLAM formulation, data association happens before computing the distance to landmarks. In our problem, however, the are no distinguishable landmarks on the boundary of the objects, except possibly for corners. Data association becomes an integral part of the optimization problem, similar to the iterative closest point algorithm [21].

Motion Cost. Terms K and L favor solutions that are consistent with models of pushing. The model we use to predict the motion of the object is based on the *limit surface*. It was proposed initially by Goyal et al. [3, 6], and later streamlined by Lynch et al. [8] and Lee and Cutkosky [7]. For simplicity of computation, we will assume a uniform pressure distribution between the object and its support. Section IV reviews the mechanics of pushing and details the construction of the motion model.

At an instant t, terms K_t and L_t represent correspondingly the difference in position and orientation between the pose $q_{t+1} = (x_{t+1}, y_{t+1}, \theta_{t+1})$ estimated by the optimizer, and the pose predicted by the motion model given the previous pose $q_t = (x_t, y_t, \theta_t)$, the estimated shape of the object $\{s_i\}_{1\dots N}$, and the observations $\mathbf{z}_t$ and $\mathbf{z}_{t+1}$.

Shape Prior Cost. We define a shape prior to prevent degenerate solutions. The term S encourages the control points $\{s_i\}_{1\ldots N}$ on the boundary of the object to be equispaced in the workspace. For point s_i, we define S_i as:

$$S_i(X) = \bar{s} - \text{dist}(s_i, s_{i+1}) \tag{2}$$

where $\bar{s} = \frac{1}{N}\sum_{i=1}^{N} \text{dist}(s_i, s_{i+1})$ and $s_{N+1} = s_1$. By minimizing $S = \sum_i S_i(X)^2$, we force all individual distances $\text{dist}(s_i, s_{i+1})$ to be closer to their mean value $\bar{s}$.

IV. The Mechanics of Frictional Planar Pushing

This section gives an overview of the mechanics of frictional pushing, and details the algebra to compute the instantaneous velocity of the pushed object $\dot{q}_t = (v_{x_t}, v_{y_t}, \omega_t)$. With an estimate for $\dot{q}$, it only remains to integrate from q_t to estimate q_{t+1}. The analysis starts from the following inputs at instant t:

- current object pose $q_t = (x_t, y_t, \theta_t)$;
- contact location $\vec{r}_{p_t}$;
- contact normal $\hat{n}_t$.
- velocity of pusher at contact $\vec{v}_{p_t}$;
- object shape $\mathcal{O}$;
- object-support pressure distribution $pd(\cdot)$;
- object-support coefficient of friction μ_s;
- object-pusher coefficient of friction μ_c.

We draw from Lynch et al. [8] and Lee and Cutkosky [7] for the following derivations.

The most important assumption we make is quasi-static interaction, i.e., accelerations do not play a decisive role. In our implementation, the exploration is performed by a contour following scheme that moves slowly, which means that the net load on the object is mostly aligned with its instantaneous velocity.

In the rest of this section we will omit the subscript t for brevity. We will use the notation $\vec{v} = (v_x, v_y)$ for the linear part of the velocity of the object, (f_x, f_y) for the linear components of the friction load and m for the angular component.

A. Limit Surface

The *limit surface* is the convex set of all possible net frictional loads (f_x, f_y, m) at the interface between a sliding object and its support surface. By virtue of the principle of maximal dissipation, each possible sliding direction induces a different frictional load. Goyal et al. [3] showed that in the case of quasi-static pushing/sliding, a particular value for a net frictional load also uniquely determines the corresponding instantaneous motion of the sliding object $\dot{q}$. Such velocity must be orthogonal to the limit surface at the point indicated by the frictional load (f_x, f_y, m).

In practice, we construct the limit surface of an object by sampling the space of possible sliding directions. A convenient approach is to sample the space of possible instantaneous rotation centers. Each rotation center $\vec{r}_{RC}$ induces a velocity on the object. This, in turn, leads to a frictional load (f_x, f_y, m), which we compute by integrating Coulomb's law over the entire support area of the object $\mathcal{O}$:

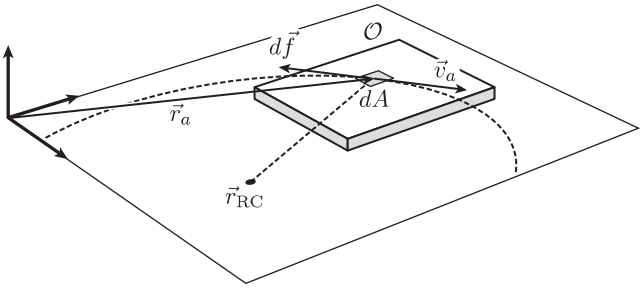

Fig. 4. Integration of infinitesimal friction force to compute the total frictional load between an object $\mathcal{O}$ and its support. If the instantaneous center of rotation of the object is at $\vec{r}_{\text{RC}}$, a point $\vec{r}_a$ with differential support dA on the support will move with velocity $\vec{v}_a$ and will contribute with an infinitesimal friction force $d\vec{f}$ that opposes $\vec{v}_a$.

$$(f_x, f_y) = -\mu_s \int_{\mathcal{O}} pd(\vec{r}_a)\frac{\vec{v}_a}{|\vec{v}_a|}dA \tag{3}$$

$$m = -\mu_s \int_{\mathcal{O}} (\vec{r}_a - \vec{r}_{RC}) \times pd(\vec{r}_a)\frac{\vec{v}_a}{|\vec{v}_a|}dA \tag{4}$$

where $\vec{r}_a$ indicates any point in the support surface, and $\vec{v}_a$ its instantaneous velocity, as illustrated in Figure 4. For a given rotation center $\vec{r}_{RC}$, we can write $\vec{v}_a = \hat{k} \times (\vec{r}_a - \vec{r}_{RC})$, with $\hat{k}$ pointing along the vertical axis.

Computing a limit surface is expensive. In our problem, since we need to search over the space of objects shapes, it becomes prohibitive. Lee and Cutkosky [7] proposed an ellipsoidal approximation that significantly speeds up its computation. We choose the center of mass of the object $\vec{r}_{\text{CM}}$ as the reference point for moments, and proceed as:

1. Find the maximum magnitude of linear friction force $f_{\text{max}} = \mu_s f_n$. This happens when the motion is a pure translation, and f_n is the total surface support force;
2. Find the maximum magnitude of torsional friction force m_{max}. This happens when the motion of the object is a pure rotation about the moment reference point. If we substitute $\vec{r}_{RC} = \vec{r}_{\text{CM}}$, and $\vec{v}_a = [0, 0, \omega]^T \times \vec{r}_a$ in (4), we obtain $m_{\text{max}} = -\mu_s \int_{\mathcal{O}} |\vec{r}_a| pd(\vec{r}_a) dA$.
3. Approximate the limit surface as an axis-aligned ellipsoid with semi-principal axis f_{max}, f_{max} and m_{max}.

$$F(f_x, f_y, m) = \left(\frac{f_x}{f_{\text{max}}}\right)^2 + \left(\frac{f_y}{f_{\text{max}}}\right)^2 + \left(\frac{m}{m_{\text{max}}}\right)^2 = 1 \tag{5}$$

The construction is illustrated in Figure 5. Lee and Cutkosky [7] and Lynch et al. [8] further discuss the validity of the approximation.

B. Motion of Object

In a quasi-static push, the magnitude of the external pushing force on an a sliding object is exactly sufficient to break friction and no greater. Therefore, it will oppose the frictional force, which we know lies on the object's limit surface. If we impose the resulting velocity $\dot{q}$ to be orthogonal to the approximated limit surface in (5), i.e., $\dot{q}$ parallel to

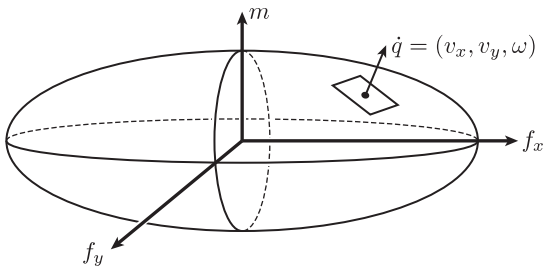

Fig. 5. Ellipsoidal approximation of the limit surface $F(f_x, f_y, m) = 0$. The lengths of the major axis of the ellipsoid are $f_{\max}$ along f_x and f_y, and $m_{\max}$ along m. Any point on the surface of the ellipsoid represents a total frictional load on a sliding object. Its resulting velocity $\dot{q}$ must be along the gradient $\nabla F(f_x, f_y, m)$.

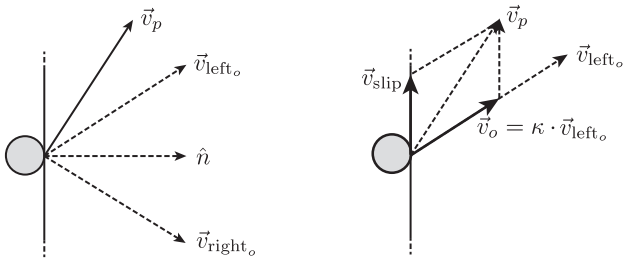

Fig. 6. (left) Probe pushing an object with a velocity $\vec{v}_p$ outside the motion cone (spanned by $\vec{v}_{\text{left}_o}$ and $\vec{v}_{\text{right}_o}$). (right) The push results into a motion to the object $\vec{v}_o$ along the left bound of the motion cone, and sliding of the pusher along the boundary of the object $\vec{v}_{\text{slip}}$.

$\nabla F(f_x, f_y, m)$, we obtain the following conditions between object force and velocity:

$$\frac{v_x}{\omega} = c^2 \frac{f_x}{m} \quad \text{and} \quad \frac{v_y}{\omega} = c^2 \frac{f_y}{m} \quad \text{with} \quad c = \frac{m_{\max}}{f_{\max}} \tag{6}$$

To relate pusher and object velocities we need to determine how the pusher motion translates into an applied force. That conversion is nicely captured by the *motion cone* introduced by Mason [5]. The following derivations are partially adapted from Lynch et al. [8].

The motion cone plays the same role for the relative velocity at a point contact interface as the friction cone plays for forces at contact. Let $\vec{v}_p$ be the velocity of the contact point on the pusher, and $\vec{v}_o$ be the velocity of the contact point on the object. The motion cone is a span of object velocities at contact such that:

- Pusher velocities $\vec{v}_p$ inside the motion cone result in the contact to *stick*, and in a perfect matching $\vec{v}_o = \vec{v}_p$.
- Pusher velocities $\vec{v}_p$ outside the motion cone result in the contact to *slide*, and in $\vec{v}_o$ lying on one of the edges of the motion cone.

The edges of the motion cone are defined by the motion of the object when the applied force is on either edge of the friction cone. In turn, the edges of the friction cone are determined by the contact normal and the object-pusher coefficient of friction μ_c. Each edge, labeled here by *left* or *right*, induces a total applied force and moment on the object. Passing those forces through (6) determines the corresponding object motions $\dot{q}_{\text{left}} = (v_{\text{left}_x}, v_{\text{left}_y}, \omega_{\text{left}})$ and $\dot{q}_{\text{right}} = (v_{\text{right}_x}, v_{\text{right}_y}, \omega_{\text{right}})$. These induce a velocity on the contact point on the object which, by definition, are the edges of the motion cone:

$$\begin{aligned} \vec{v}_{\text{left}_o} &= (v_{\text{left}_x}, v_{\text{left}_y}) + \omega_{\text{left}} \cdot \hat{k} \times \vec{r}_o \\ \vec{v}_{\text{right}_o} &= (v_{\text{right}_x}, v_{\text{right}_y}) + \omega_{\text{right}} \cdot \hat{k} \times \vec{r}_o \end{aligned} \tag{7}$$

where $\vec{r}_o$ is defined such that it indicates the location of contact with respect to the origin of moments. We proceed with the analysis independently for cases where the contact sticks or slides.

Case I: Pusher sticks to the object. When the pusher motion is inside the motion cone, the contact sticks and $\vec{v}_o = \vec{v}_p$. Abusing vector notation, we express $\vec{v}_o$ as a function of the object generalized velocity $\dot{q} = (\vec{v}, \omega)$ as $\vec{v}_o = \vec{v} + \omega \cdot \hat{k} \times \vec{r}_o$. The condition turns into:

$$\begin{aligned} v_x - \omega \cdot r_{o_y} &= v_{p_x} \\ v_y - \omega \cdot r_{o_x} &= v_{p_y} \end{aligned} \tag{8}$$

Additionally, the moment and force generated on the object are related by $m = \vec{r}_o \times (f_x, f_y, 0)_z$, which expands as $m = r_{o_x} f_y - r_{o_y} f_x$. We solve equations (6) and (8) as:

$$\begin{aligned} v_x &= \frac{(c^2 + r_{o_x}^2) v_{p_x} + r_{o_x} r_{o_y} v_{p_y}}{c^2 + r_{o_x}^2 + r_{o_y}^2} \\ v_y &= \frac{r_{o_x} r_{o_y} v_{p_x} + (c^2 + r_{o_y}^2) v_{p_y}}{c^2 + r_{o_x}^2 + r_{o_y}^2} \\ \omega &= \frac{r_{o_x} v_y - r_{o_y} v_x}{c^2} \end{aligned} \tag{9}$$

Note that interestingly, the resulting velocity is independent of the object-support coefficient of friction μ_s, a known phenomenon for quasi-static pushing.

Case II: Pusher slides on the object. When the pusher motion is outside the motion cone the contact slides and $\vec{v}_o$ lies on the boundary of the motion cone, in the direction or either $\vec{v}_{\text{left}_o}$ or $\vec{v}_{\text{right}_o}$. Let $\vec{v}_b$ be the choice of boundary.

Only partial motion of the pusher is transferred to the object. Part is lost in slipping. The fraction that is transferred can be written as $\vec{v}_o = \kappa \vec{v}_b$, where $\kappa = \frac{\vec{v}_p \cdot \hat{n}}{\vec{v}_b \cdot \hat{n}}$ is the factor that subtracts the sliding velocity at contact, as illustrated in Figure 6. The instantaneous velocity of the object can be recovered now with (9) as if it were an "effective" pusher that sticks to the object with pushing velocity $\vec{v}_o$.

V. Experiments

In this paper we evaluate the proposed approach in 2D with simulated data. All the following experiments are conducted in MATLAB on an Intel i7 desktop PC with 16GB memory.

The experiments use a contour following algorithm as the tactile exploratory procedure. Lederman and Klatzky [22] categorized the different exploratory procedures that humans use to recover object properties such as identity, texture, weight or hardness, and observed that contour following is

the most effective way to obtain an accurate global shape. These experiments show that the stream of tactile data originating from such an exploration policy is sufficient to recover the shape and trajectory of the object, even in the presence of noise.

A. Simulated dataset

In our implementation, a square object of 100 mm in side is pushed by a circular probe of 5 mm in radius. The probe pushes the object while following its contour. The motion of the probe is controlled based on the previous motion direction and the current contact normal.

We simulate planar pushing as described in Section IV. Friction coefficients are: $\mu_s = \mu_c = 0.25$. The pusher starts close to the bottom left corner and in contact with the object. The dataset is comprised of $10,000$ simulated steps for up to 4 complete turns around the contour of the object. Figure 2a and Figure 2b show the contact points and contact normals collected from the simulation.

B. Shape and Pose recovery

For the experiments in this paper, we do a $\frac{1}{25}$ subsampling of the dataset, for a total of $T = 400$ timestamps. We input the following values:

- The number of control points for the shape is $N = 8$.
- The initial guess for the shape is roughly a circular polygon centered at the origin and with a radius of 30 mm, smaller than the true shape, as illustrated in Figure 2c.
- The initial guess for the trajectory of the object, detailed in Appendix A, is a less trivial approximation based on the automatic detection of corners.
- The weights $\boldsymbol{\alpha}$ of the cost function are experimentally set to $[10, 0.5, 1, 10, 1]^T$.

We optimize the cost function (1) with the Matlab function `nonlinsq()`, which uses a *trust-region-reflective* method. The optimization takes in the order of 40 minutes to converge. Figure 2c and Figure 2d shows a snapshot of the starting initial guess, and the final optimized solution for one time frame t. Figure 7 shows a qualitative comparison with ground truth over more time steps.

To evaluate the performance of the algorithm, we define a metric $\mathcal{G}$ that, for a particular solution $\hat{X} = (\{(x_t, y_t, \theta_t)\}_{1\dots T}, \{s_i\}_{1\dots N})$ to Problem 1, averages the minimum distance from each shape control point to the true boundary of the object $\partial\mathcal{O}$. More precisely, if H_t is the 2D transformation induced by the true pose of the object at time t, and $\hat{H}_t$ the estimated one given by (x_t, y_t, θ_t):

$$\mathcal{G}(\hat{X}) = \sqrt{\frac{1}{NT}\sum_{t=1}^{T}\sum_{i=1}^{N} \text{min-dist}(\hat{H}_t(s_i), H_t(\partial\mathcal{O}))^2} \quad (10)$$

where min-dist computes the minimum distance from a point to a shape, in our case a polygon. Intuitively, $\mathcal{G}(\cdot)$ measures how much each shape control point is off from the true object shape, averaged over time and control points.

TABLE I
EXPERIMENTAL RESULTS

Experiment	$\mathcal{G}$ (mm)	Resnorm	Time (10^3s)
P+S	7.1	6.5	1.9
P	5.3	0.0	2.5
S	4.6	1.6	2.4
P+S-D	7.6	0.0	2.5
P+S-G	9.0	17.1	2.4
P+S-K	9.8	20.5	2.3
P+S-L	6.5	0.0	2.1
P+S-S	7.5	0.0	2.5

Columns left to right: Type of experiment, final value of the metric $\mathcal{G}$, final value of the cost function, and computation time.

To evaluate the performance of the algorithm, we conducted the following experiments:

- P Find the trajectory of poses of the object $\{(x_t, y_t, \theta_t)\}_{1\dots T}$ assuming its shape is known.
- S Find the shape of the object $\{s_i\}_{1\dots N}$ assuming its trajectory of poses is known.
- P+S Find both the shape and the trajectory of the object.
- P+S-x In this experiment, we analyze the relevance of each term in the cost function $E(X)$ to the solution of P+S. We individually set their weights to zero. The term x indexes one of $\{D, G, K, L, S\}$.
- P+S+σ In this experiment, we analyze the sensitivity of the solution of P+S in the presence of Gaussian noise in both the observation and motion models.

The results are discussed in the following section.

VI. DISCUSSION

A. Results

The top rows of Table I show that, as expected, the error lowers when we provide information to the optimizer in the form of either the ground truth for shape or pose. The results also reflect that it is relatively easier to recover the shape than the trajectory of the object. However, even with perfect information for the pose of the object, we get some error in the estimation of its shape, which we attribute to the downsampling of the data.

We discuss now the respective contribution of the different factors that compose the cost function. The experiments P+S-x in Table I reflect that:

- Terms G and K seem to help the optimization better than the other terms, since both the cost function and $\mathcal{G}$ increase, which could indicate a local minimum.
- When either of the terms D, L, or S are removed from the optimization, the cost function reaches zero without $\mathcal{G}$ being zero, which could indicate that the problem becomes under-constrained.
- Note that the term L, which captures the rotational component of the motion prediction, seems to contribute negatively to the optimization. We can only speculate here that the rotation prediction from the model might

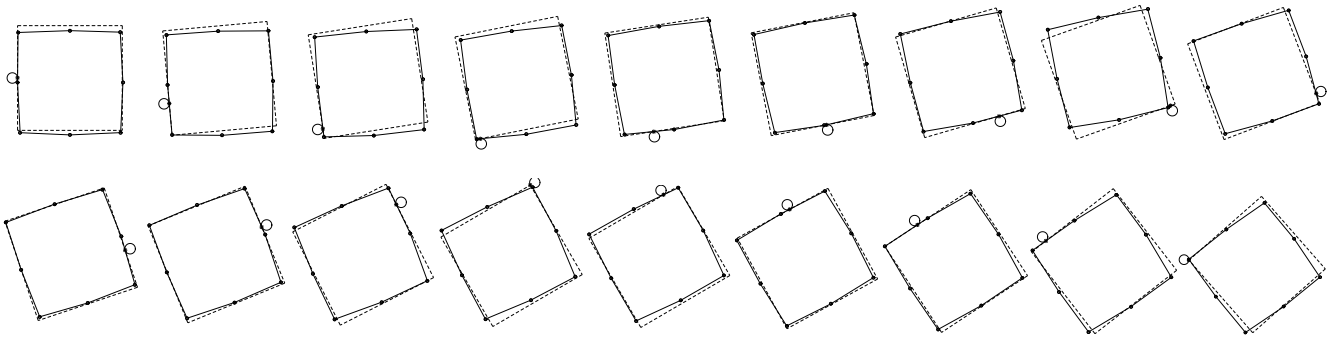

Fig. 7. Comparison of recovered pose and shape (solid line) with ground-truth (dashed line). Time steps from left to right and top to bottom. The circle indicates the pusher. The knots on the solid line indicate the vertices of the estimated shape.

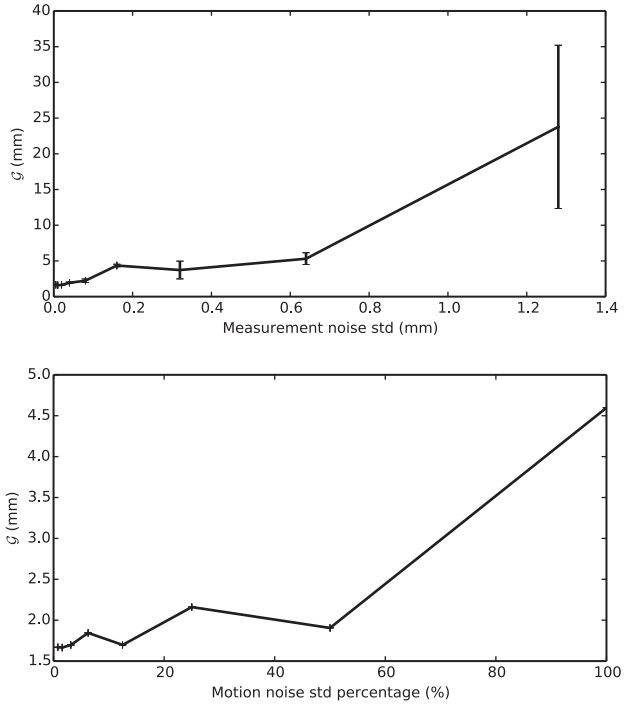

Fig. 8. Sensitivity of the optimization to Gaussian noise in the observation (contact location) and motion models. (top) The observation noise is additive white noise in the contact location. (bottom) The motion noise is introduced as a percentage with respect to the predicted in one step of simulation. As expected, the metric $\mathcal{G}$ grows with the noise.

not be too informative compared to the initial guess (derived without downsampling).

Finally Figure 8 shows the sensitivity of the solution to the presence of disturbances in the motion and observation models. We conduct the P+S experiment where we add different magnitudes of noise to the predicted instantaneous motion of the object and the observed contact location.

Overall the experiments support the hypothesis that tactile information, along with knowledge of the mechanics of pushing, is sufficient to recover the trajectory and pose of a movable planar object.

B. Limitations

The first and foremost limitation of the system is that we assume a known uniform pressure distribution. This, given the contact location and along with a quasi-static assumption on the interaction between the object and the support surface, gives a deterministic one-to-one mapping between pusher velocities and predicted object motions. The reality of friction however is that it more difficult to predict. Small changes to the pressure distribution can change the instantaneous center of rotation of the object. A more mature implementation could use bounds for that uncertainty.

We also assume Gaussian noise in the contact measurements which makes it suitable for a least-squares formulation. But contact is essentially non-Gaussian. When there is contact between two objects, the set of relative feasible poses is a lower dimensional manifold that cannot be nicely represented by a Gaussian distribution. Koval et al. [23] formalize the concept of particle manifold filter that respects that lower dimensionality, which could lead to more robust implementations of the estimation of shape and pose.

Finally, the algorithm uses a batch optimization, but a more efficient exploration would use current estimations to guide future tactile motions. This would require an on-line version of our algorithm.

VII. Conclusion and Future Work

Motivated by a human's ability to blindly explore objects by touch, we present a method to recover the shape and trajectory of a movable 2D object, while pushed by a probe. We show that both can be recovered from a stream of contact locations and contact normals, by exploiting knowledge of the mechanics of pushing.

We contribute with a detailed formulation of the problem, an approach to solve it, a metric to evaluate the fitness of a solution, and an experimental study of the performance of the approach on simulation data.

The approach defines a least-squares cost function that fits the data to observation and motion models, along with a prior on the representation of the shape. We have tested the approach on a simulated dataset under controlled noise to study the solvability and sensitivity of the problem.

Future research will include experiments with a real robot to evaluate the system performance for varying shapes, weights, materials and roughness. In addition, we will investigate on-line estimation of shape and pose, in the style of the iSAM framework by Kaess et al. [24], to achieve real-time active tactile exploration.

APPENDIX
INITIAL GUESS FOR POSES

The initial guess for the shape of the object is straightforward, with the control points $\{s_i\}_{1\dots N}$ placed in a small circular configuration. If we start the optimization, however, with a trivial value for the trajectory of the pose $\{(x_t, y_t, \alpha_t)\}_{1\dots T}$, the optimization does not converge.

The initial guess for the trajectory is relatively more important than that for shape. Observed contact points assigned to edges too far from ground truth in the observation model can guide the optimization in a very wrong direction. The orientation seems to be particularly relevant for that purpose. Our initial guess for the rotational trajectory of the object comes from an automatic detection of corners on the object. These give us a chance to estimate how much rotation is performed during a complete cycle of contour following. The key steps are as follows:

1) Compute the second derivative with respect to t of the pusher-object contact location. Since the object moves slowly, we can regard it as an approximation of the curvature of the object.
2) Find peaks in the "curvature" profile, as in Figure 9. Use a window-based approach to suppress erroneous multiple peaks. Denote the set of monotonically increasing peaks by $\{g\}_{1\dots G}$.
3) Make guesses on the true number of corners of the object. Each guess leads to a different interpretation for the sequence of peaks. Score each value of the number of corners by the corresponding periodicity of the distances between peaks, and chose the best fit.
4) Estimate the average period ΔT of the number of samples that it takes for one complete revolution of the probe around the object.
5) Estimate the angle the object rotates $\Delta\theta$ in one full revolution of the probe along the object $\hat{n}_t$ and $\hat{n}_{t+\Delta T}$.
6) Set the initial guess for the rotation as the linear interpolation $\theta_t = t \cdot \frac{\Delta\theta}{\Delta T}$.

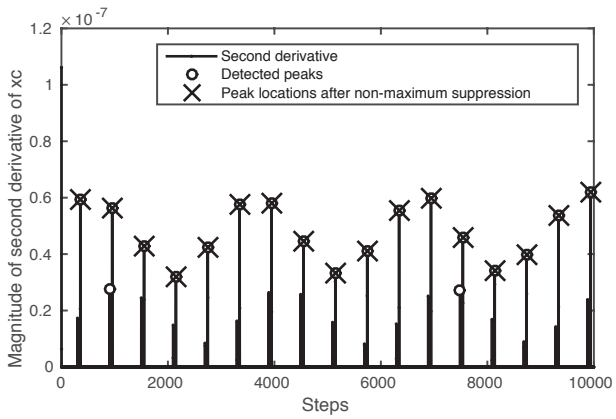

Fig. 9. Numerical approximation of the second derivative of the detected contact points. The crosses are detected as peaks, which correspond to the corners of the shape. The periodicity of the profile corresponds to different revolutions of the probe along the object.

REFERENCES

[1] L. Jones and S. Lederman, "Chapter 4: Tactile sensing," in *Human Hand Function*. Oxford University Press, 2006, pp. 44–74.

[2] R. L. Klatzky, S. J. Lederman, and V. A. Metzger, "Identifying objects by touch: An expert system," *Perception & Psychophysics*, vol. 37, no. 4, pp. 299–302, 1985.

[3] S. Goyal, A. Ruina, and J. Papadopoulos, "Planar Sliding with Dry Friction Part 1. Limit Surface and Moment Function," *Wear*, vol. 143, pp. 307–330, 1991.

[4] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[5] M. T. Mason, "Mechanics and planning of manipulator pushing operations," *IJRR*, vol. 5, no. 3, pp. 53–71, 1986.

[6] S. Goyal, A. Ruina, and J. Papadopoulos, "Planar Sliding with Dry Friction Part 2. Dynamics of Motion," *Wear*, vol. 143, pp. 331–352, 1991.

[7] S. H. Lee and M. Cutkosky, "Fixture planning with friction," *Journal of Manufacturing Science and Engineering*, vol. 113, no. 3, pp. 320–327, 1991.

[8] K. M. Lynch, H. Maekawa, and K. Tanie, "Manipulation and active sensing by pushing using tactile feedback." in *IROS*, 1992, pp. 416–421.

[9] M. Moll and M. A. Erdmann, "Reconstructing the shape and motion of unknown objects with active tactile sensors," in *Algorithmic Foundations of Robotics V*. Springer, 2004, pp. 293–310.

[10] Y.-B. Jia and M. Erdmann, "Pose and motion from contact," *IJRR*, vol. 18, no. 5, pp. 466–487, 1999.

[11] A. Petrovskaya and O. Khatib, "Global localization of objects via touch," *IEEE TRO*, vol. 27, no. 3, pp. 569–585, 2011.

[12] U. Martinez-Hernandez, G. Metta, T. J. Dodd, T. J. Prescott, L. Natale, and N. F. Lepora, "Active contour following to explore object shape with robot touch," in *World Haptics Conference*, 2013, pp. 341–346.

[13] C. Strub, F. Worgotter, H. Ritter, and Y. Sandamirskaya, "Using haptics to extract object shape from rotational manipulations," in *IEEE/RSJ IROS*, Sept 2014, pp. 2179–2186.

[14] L. Zhang, S. Lyu, and J. Trinkle, "A dynamic Bayesian approach to real-time estimation and filtering in grasp acquisition," in *IEEE ICRA*, 2013, pp. 85–92.

[15] M. Koval, N. Pollard, and S. Srinivasa, "Pose estimation for planar contact manipulation with manifold particle filters," *International Journal of Robotics Research*, vol. 34, no. 7, pp. 922–945, June 2015.

[16] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[17] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski, "Building Rome in a day," *Communications of the ACM*, vol. 54, no. 10, pp. 105–112, 2011.

[18] S. Thrun and M. Montemerlo, "The GraphSLAM algorithm with applications to large-scale mapping of urban structures," *IJRR*, vol. 25, no. 5/6, pp. 403–430, 2005.

[19] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. Leonard, and J. McDonald, "Kintinuous: Spatially extended KinectFusion," in *RSS Workshop on RGB-D*, Sydney, Australia, Jul 2012.

[20] F. Dellaert and M. Kaess, "Square Root SAM: Simultaneous localization and mapping via square root information smoothing," *IJRR*, vol. 25, no. 12, pp. 1181–1204, Dec. 2006.

[21] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," *IEEE PAMI*, 1992.

[22] S. J. Lederman and R. L. Klatzky, "Hand movements: A window into haptic object recognition," *Cognitive psychology*, vol. 19, no. 3, pp. 342–368, 1987.

[23] M. Koval, M. Dogar, N. Pollard, and S. Srinivasa, "Pose estimation for contact manipulation with manifold particle filters," in *IEEE/RSJ IROS*, 2013, pp. 4541–4548.

[24] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *IEEE TRO*, vol. 24, no. 6, pp. 1365–1378, Dec. 2008.